EE4302 Advanced Control System CA1

A0260014Y
Guo Shiping

Contents

1. Review of State-Variables and MATLAB

$$\dot{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}} = \begin{bmatrix} 0 & 1 \\ -1.2 & -3.71 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$F = A = \begin{bmatrix} 0 & 1 \\ -1.2 & -3.71 \end{bmatrix}; \; G = B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}; \; H = C = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$u = -k_1 x_1 - k_2 x_2 = 0.1x_1 - 0.1x_2$$

Open loop bode diagram and step response:

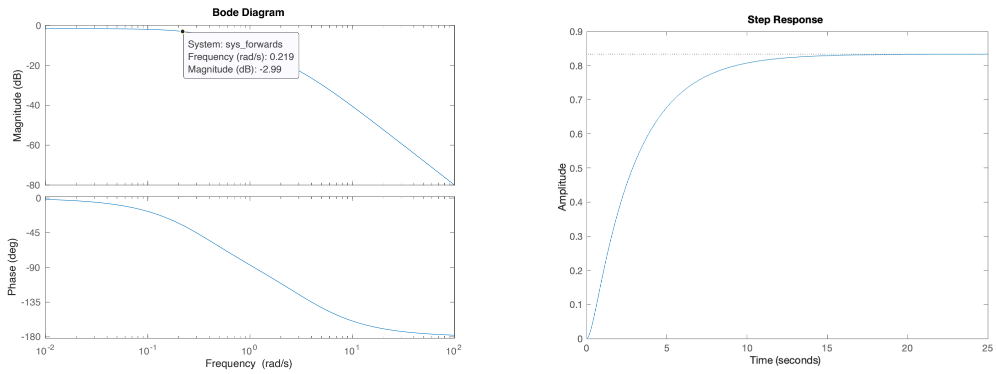

Figure 1 Open loop bode diagram and step response

There are two approaches in MATLAB to simulate the feedback control, one solution is using feedback function and another is use ss function to structure the close loop system. I will try both of them.

1.1 Use Feedback function

```
sys_forwards = ss(A_Plant, B_Plant, C_Plant_full, D_Plant); % forwards state space
sys_backwards = ss(0, [0, 0], 0, [0.1, 0.1]);   % backwards state space
sys_feedback = feedback(sys_forwards, sys_backwards, -1); % feedback control
sys_feedback = ss(sys_feedback.A, sys_feedback.B, [1,0,0], 0); % output y = Hx
scaling_fb_gain = 1/dcgain(sys_feedback);
sys_feedback = Scaling_fb_gain*sys_feedback;
bode(sys_feedback)
```

Thinking:
The backwards state space form is a trap.
The input and output relationship of backwards feedback should be

$$\begin{cases} u = -k_1 x_1 - k_2 x_2 = 0.1x_1 - 0.1x_2 \\ y = Ku = \begin{bmatrix} 0.1 & 0.1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \end{cases}$$

To achieve this, initiate the feedback process before the multiplication with H. Once the closed-loop system is obtained, rearrange the output by multiplying it with HH to obtain the final output. Additionally, note that the step response steady state is below 1, indicating a steady-state gain less than 1. To rectify this and attain a steady-state gain of 1, incorporate a scaling gain by multiplication.

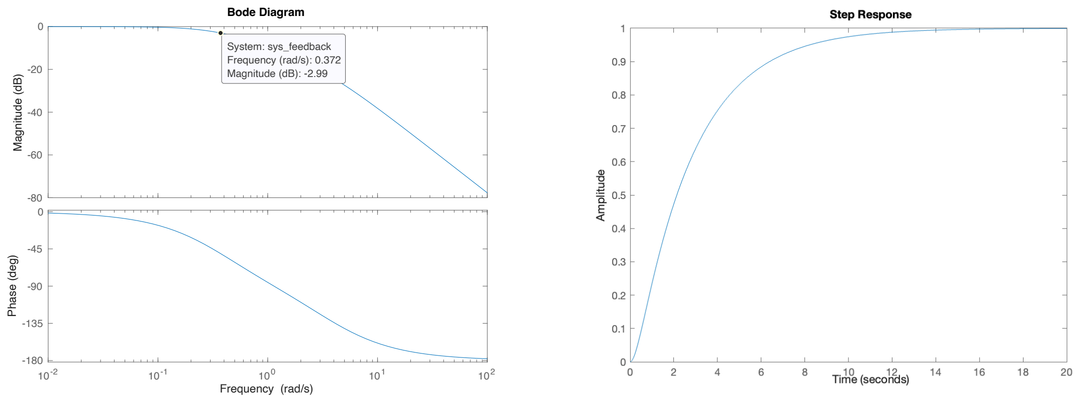

Figure 2 Close loop bode diagram and step response by using feedback function

1.2 Not use feedback function to check the result.

```
K = [0.1, 0.1];
sys_fb_check = ss(A_Plant - B_Plant*K, [0; 1], C_Plant, D_Plant); % State spcce formal
DC_fb_gain = 1/dcgain(sys_fb_check);
sys_fb_check = DC_fb_gain*sys_fb_check;
step(sys_fb_check)
```

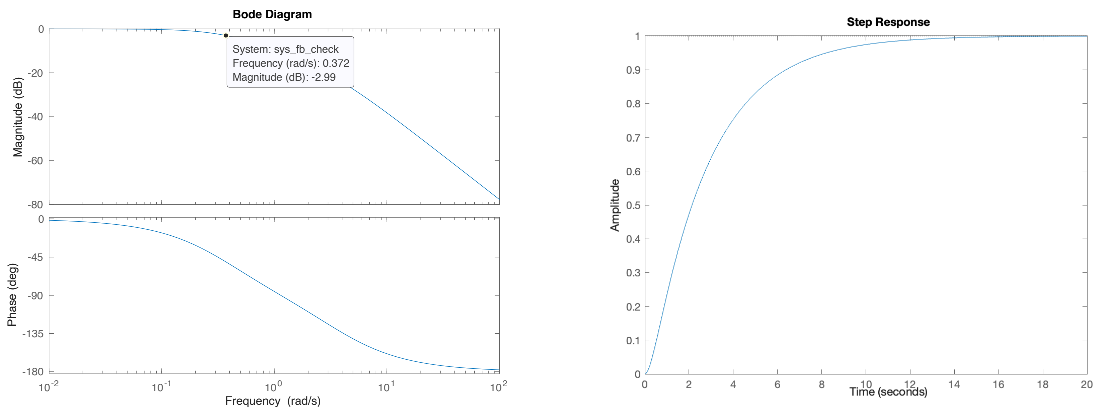

Figure 3 Close loop bode diagram and step response by using ss function

These two methods results are the same, so I am confident that my simulations are right.

1.3 Thinking:

For the command below

$$\dot{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}} = \begin{bmatrix} 0 & 1 \\ -1.2 & -3.71 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0.1 & 0.1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ r \end{bmatrix}$$

```
sys_fb_check = ss(A_Plant - B_Plant*K,  [0; 1], C_Plant, D_Plant); % State spcce formal
```

Need set B as [0; 1], which is the reference signal or steady state input.

So the input signal should be written as

$$u = \begin{bmatrix} 0.1 & 0.1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + r$$

It caters for the flow chart in figure 1. This is a direct way to see the structure.

2. Simple State-Feedback Design [CA1- 20 module marks]

2.1 Design Using Ackermann's Formula

Based on the closed-loop specifications given above, and the relation between location of poles and bandwidth, choose your desired closed-loop pole locations using the ITAE criterion. Then using Ackermann's formula, calculate the state-feedback gains to place the poles of the closed-loop at the desired locations.

Closed-loop bandwidth:	Not lower than 3.5 rad/s;
Resonant Peak, M_r:	Not larger than 2dB;
Steady-state gain between r and y:	0dB.

Provide a listing of how you used MATLAB to do the computations in the Ackermann's formula, and also results of the computations.

Provide calculations to show how the scaling gain was chosen.

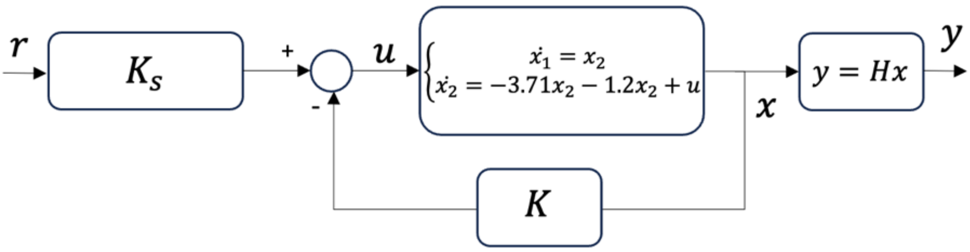

Figure 4 Close loop system structure

The process to calculate using Ackermann's formula:

First, we use Second-Order Dominant Response methodology.

Based on the requirement we can calculate the reference model.

$$\begin{cases} \hat{G}(s) = \frac{w_n^2}{s^2+2\xi w_n + w_n^2} \\ W_b = w_n\sqrt{1-2\xi^2+\sqrt{2-4\xi^2+4\xi^4}} > 3.5 \\ W_r = w_n\sqrt{1-2\xi^2} \end{cases}$$

Calculate by Matlab:

```
syms sgm wn s
eqn1 = abs(sgm) < 1.8478;
```

```
eqn2 = abs(sgm) > 0.7654;
eqn3 = wn*(sqrt(1-2*sgm^2+sqrt(2-4*sgm^2+4*sgm^4))) > 3.5;
conds = [eqn1, eqn2, eqn3];
sol = solve(conds, [wn, sgm], 'ReturnConditions', true)
```

Calculating by hand:

$$\frac{2-\sqrt{2}}{2} < \xi^2 < \frac{2+\sqrt{2}}{2} \Rightarrow \begin{cases} -1.8478 < \xi < -0.7654 \\ 1.8478 > \xi > 0.7654 \end{cases}$$

let

$$\xi = 0.8 \Rightarrow w_n = 4.0188$$

Show the calculation process :

```
syms s
sgm = 0.8;
wn = 4.0188;
eqn3 = s^2 + 2*sgm*wn*s + wn^2 == 0;
S = solve(eqn3)
```

S =

$$\begin{pmatrix} -\frac{10047}{3125} - \frac{\sqrt{1598214708416090569277 1}\ \mathrm{i}}{52428800000} \\ -\frac{10047}{3125} + \frac{\sqrt{1598214708416090569277 1}\ \mathrm{i}}{52428800000} \end{pmatrix}$$

So the two poles are

```
ans =

  -3.2150 - 2.4113i


ans =

  -3.2150 + 2.4113i
```

Open loop:

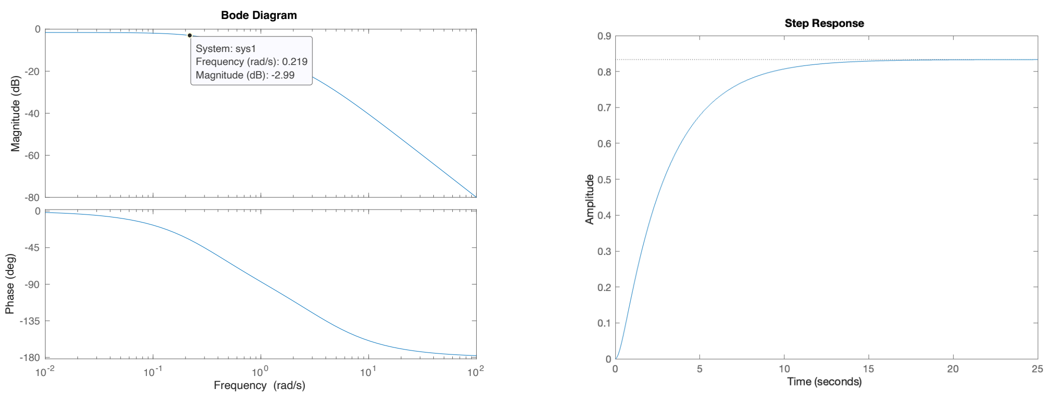

Figure 5 Open loop bode diagram and step response

```
Poles =

   -0.3580
   -3.3520
```

Obviously, it doesn't satisfy the requirement.

Is it controllable?

*Controllability_Matrix = [B_Plant, A_Plant*B_Plant];*
rank(Controllability_Matrix) = 2

So it's controllable.

Next, we use Ackermann's formula to control this system.

1. We need the target poles. Here are three methods to get the target poles.

ITAE and Bessel prototype table methodology, Second-Order Dominant Response methodology

ITAE and Bessel methods are based on known table and similar.

Use Ackermann's formula procedure:

1. Need the target poles and get.

$$\alpha_c(s) = s^n + \alpha_1 s^{n-1} + \ldots + \alpha_n = 0$$

$$\alpha_c(s) = s^2 + 6.43s + 16.1506$$

2. Calculate the $a_c[F]$

$$\alpha_c(F) = F^n + \alpha_1 F^{n-1} + \ldots + \alpha_n I$$

$$\alpha_c(F) = F^2 + 6.43F + 16.1506I$$

$$= \begin{bmatrix} 0 & 1 \\ -1.2 & -3.71 \end{bmatrix}^2 + 6.43\begin{bmatrix} 0 & 1 \\ -1.2 & -3.71 \end{bmatrix} + \begin{bmatrix} 16.1506 & 0 \\ 0 & 16.1506 \end{bmatrix}$$

3. Calculate the ζ^{-1}

$$\zeta = \begin{bmatrix} G & FG & \ldots & F^{n-1}G \end{bmatrix}$$

$$\zeta = [G \quad FG] = \begin{bmatrix} 0 & 1 \\ 1 & -3.71 \end{bmatrix}$$

$$rank(\zeta) = 2$$

4. The feedback law

$$u = -Kx = -[0 \quad \dots \quad 0 \quad 1]\zeta^{-1}\alpha_c(F)x$$

$$K = [01]\zeta^{-1}\alpha_c(F) = \begin{bmatrix}14.9506 & 2.72\end{bmatrix}$$

So controllability matrix ζ^{-1} must be non-singular we can check it first before using Ackermann's formula.

Now I got the feedback K, directly use this K without scaling gain we can get the bode diagram and step response as follows:

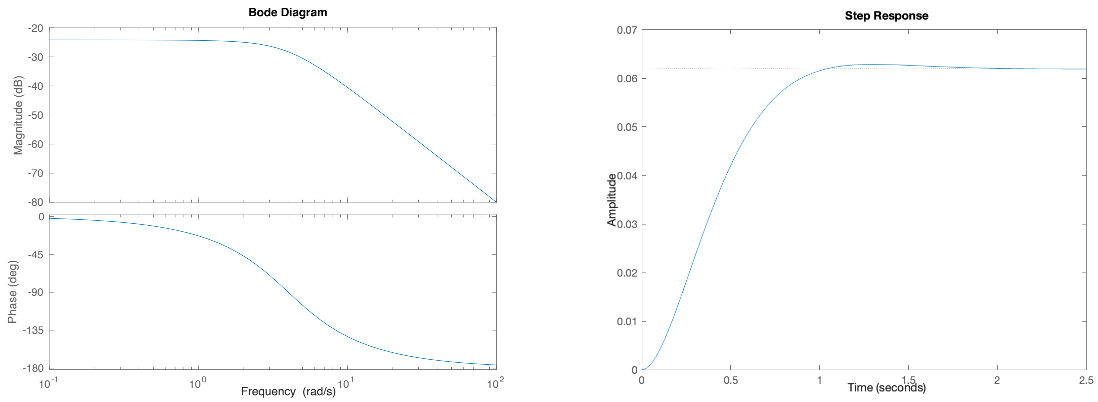

Figure 6 SOD approach bode diagram and step response without using scaling gain

It doesn't satisfy the requirement !

Scaling gain calculation:

With the feedback K we can easily get the closed state space model of this system.

$$\begin{bmatrix}\dot{x}_1 \\ x_2\end{bmatrix} = \begin{bmatrix}0 & 1 \\ -16.15 & -6.43\end{bmatrix}\begin{bmatrix}x_1 \\ x_2\end{bmatrix} + \begin{bmatrix}0 \\ 1\end{bmatrix}u$$

Then transfer state space into transfer function:

$$G = \frac{1}{s^2 + 6.43s + 16.1506}$$

Let s = 0 we can get the steady gain of the closed system.

$$G(0) = \frac{1}{16.1506}$$

The scaling gain is 16.1506.

Next step multiple scaling gain into the close system we can get the new closed system, and the bode diagram step response are following.

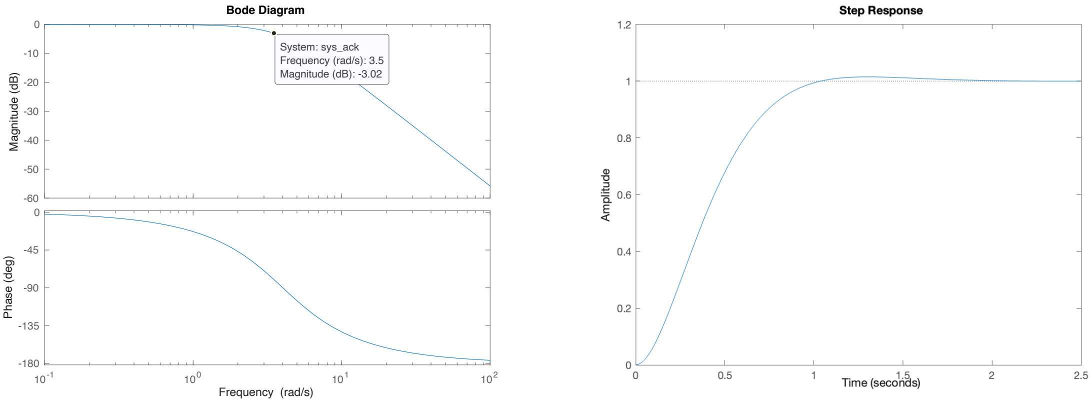

Figure 7 SOD approach bode diagram and step response with scaling gain

Now, the result satisfies the requirement.

We can also use ITAE Design, the procedure is the same as Second-Order Dominant Response methodology. The only different is how to choose the desired poles.

Compared with Second-order reference model approach, this method is much easier. We can directly get the target poles by looking up the table.

```
ITAEPoles = 3.5*[-0.7071+0.7071*1i; -0.7071-0.7071*1i];      % ITAE Design Poles as a column vector
```

All the other calculation are the same with Second-order reference model approach.

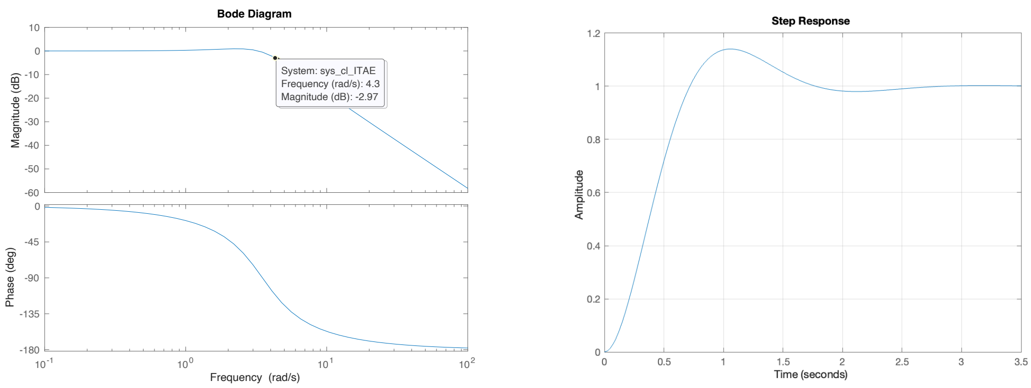

Figure 8 ITAE approach bode diagram and step response

ITAE Design Poles scaling gain: 12.2498

Bessel prototype table methodology:

```
tf_ITAE =

               s
  ------------------------
  s^3 + 3.71 s^2 + 12.25 s
```

Bessel Prototype

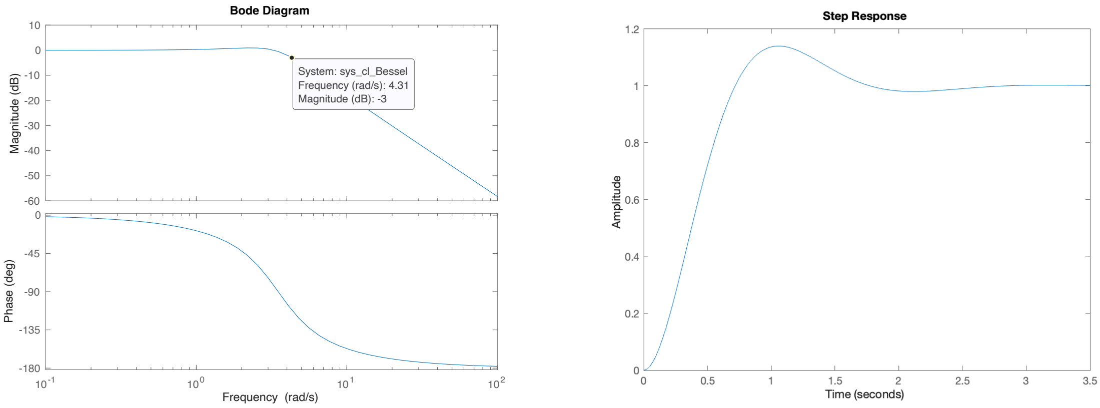

Figure 9 Bessel prototype approach bode diagram and step response

Ks_Bessel = 12.2495

These three approaches can achieve the goal very well.

3. LQR approach

Here I try to find the systemic procedure to solve LQR controller.
The LQR optimal control is to find the control law that minimize the cost function:

$$J = \frac{1}{2}\int_0^{\infty}(x^T Q x + u^T R u)dt$$

It turns out to be in the form of linear state feedback: u=r-Kx.

First is to find the positive definite solution P of the Riccati quuation:

Where A and B are plant parameters, Q and R are parameters given in LQR method.

It's the key process to solve LQR problem. It will generate many non-linear equations so it's hard to solve it. For singular problem we can directly solve the equation, but for matrix problem here we have systemic steps to solve ARE:
Try

$$Q = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$$

$$R = 1$$

We have 3 parameters can be changed.

Step1: Form the matrix

$$\tau = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix}$$

```
Tau = 4x4
         0    1.0000         0         0
   -1.2000   -3.7100         0   -1.0000
  -10.0000         0         0    1.2000
         0  -10.0000   -1.0000    3.7100
```

Step2: Separate the selected stable eigenvectors(corresponding to stable eigenvalues) matrix to two parts:

```
[V, D] = eig(Tau)

V = 4x4
   -0.0243   -0.0470   -0.0835    0.1289
   -0.1109   -0.0348    0.0619   -0.5882
    0.3022    0.9757   -0.9911    0.4551
    0.9465    0.2114   -0.0836   -0.6560

D = 4x4
    4.5623         0         0         0
         0    0.7414         0         0
         0         0   -0.7414         0
         0         0         0   -4.5623
```

There are two negative eigenvalues.
Corresponding vectors are

	1	2	3	4
1	-0.0243	-0.0470	-0.0835	0.1289
2	-0.1109	-0.0348	0.0619	-0.5882
3	0.3022	0.9757	-0.9911	0.4551
4	0.9465	0.2114	-0.0836	-0.6560

$$\begin{bmatrix} v_i \\ \mu_i \end{bmatrix}, i = 1, 2.$$

Step3: P is given by:

$$Pv_i = u_i, \quad i = 1, 2$$

Then we can get the feedback K.

$$K = R^{-1}B^T P$$

We got the final K as:

```
v = V(1:2,3:4);
u = V(3:4,3:4);
P = u*inv(v);
K = inv(R)*G'*P
```

```
K = 1x2
      2.1823    1.5937
```

Check the result by using the lqr function in MATLAB.

```
LQR_K = lqr(F,G, Q,R)
```

```
LQR_K = 1x2
      2.1823    1.5937
```

The results are the same.

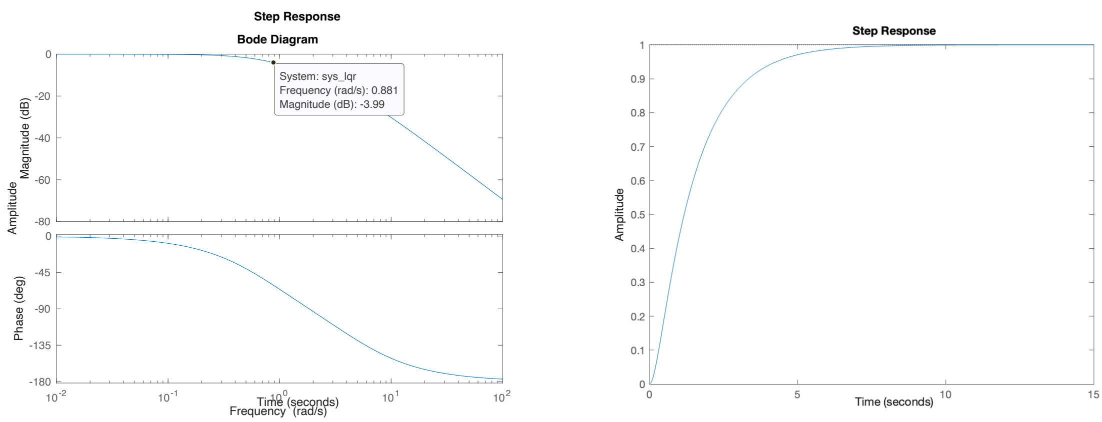

Figure 10 LQR approach bode diagram and step response

Mainly show how to choose Q and R.

By experimenting with different values for Q and R, I observed that increasing Q1 enhances the frequency bandwidth. However, the second parameter of Q has a negative impact on the frequency bandwidth. Therefore, I maintained consistency in R and adjusted the two parameters in Q accordingly. After numerous tests, I arrived at the final selection of Q and R as follows.

Q_LQR = [320, 0; 0, 1]; % Define penalty matrix Q
R_LQR = 1; % Define penalty matrix R (in this case, a scalar)

The result can satisfy the requirement.

The following steps are the same with other approaches.

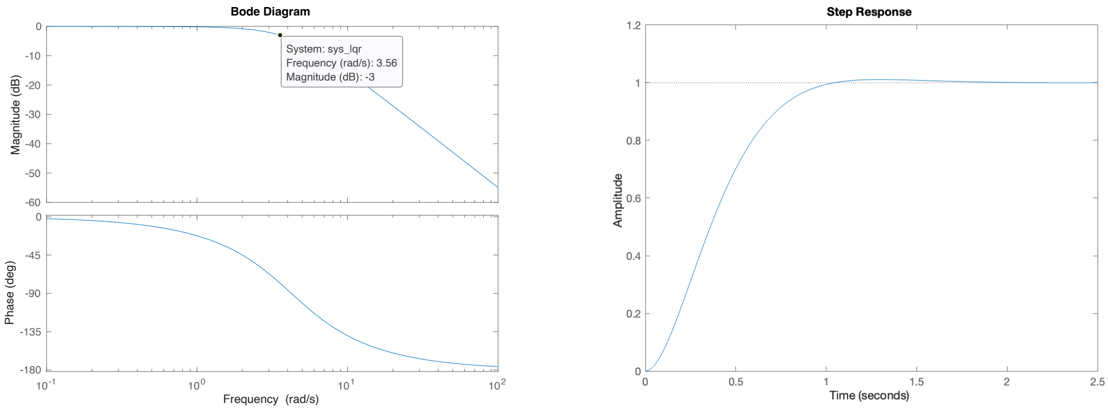

Figure 11 LQR approach bode diagram and step response with satisfied parameters

4. Conclusion

In conclusion, this experiment delves into advanced control systems, emphasizing the significance of state-variable analysis and MATLAB simulations. The exploration of open-loop bode diagrams and step responses, along with a critical understanding of feedback functions, unveils essential considerations in system modelling.

The meticulous approach to state-feedback design using Ackermann's Formula and ITAE criterion, coupled with MATLAB computations and scaling gain considerations, demonstrates a comprehensive understanding of the design process.

The exploration of LQR controllers further showcases a systematic approach to control problem-solving. The careful selection of Q and R matrices, considering their impact on frequency bandwidth, solidifies the report's commitment to a holistic understanding of advanced control systems.

Most important, it helps me understand the theory learnt in class and prepare for the final exam.